ELSEVIER

Discrete Applied Mathematics 53 (1994) 291–298

DISCRETE APPLIED MATHEMATICS

Practical experiments of broadcasting algorithms on a configurable parallel computer†

Philippe Michallon, Denis Trystram*

LMC-IMAG, 46, Avenue Félix Viallet, 38031 Grenoble Cedex, France

Received 16 September 1992; revised 22 June 1992

Abstract

The goal of this paper is to present practical experiments on broadcasting algorithms on a configurable parallel computer based on *Transputers*. We present briefly the communication model of this architecture. Then, we develop some broadcasting algorithms and give experimental results.

Key words: Broadcasting; Store and forward communication model; Full-duplex communications; Arc-disjoint spanning trees; Configurable parallel architectures

1. Introduction

During the last decade, the number of commercially available distributed-memory parallel computers has grown considerably. The researchers have shown the great interest of using this kind of computers for accelerating the execution time of algorithms. Many of them (Gaussian elimination, iterative systems solvers, etc.) use global communication schemes like the broadcast of data. Most practical implementations of such algorithms have been proposed on static regular networks of processors (namely, hypercube, grid, ring, trees, etc.) [6, 12, 14].

This paper deals with the analysis of the communications of a real-life massively parallel machine in order to propose an efficient practical broadcasting routine. After having recalled the broadcasting problem, we present briefly a modelization of basic communications on a MIMD parallel computer. This analysis emphasizes that the communication time is greatly influenced by the number of links used in parallel. Then, we propose various strategies for broadcasting and we compare their implementations.

† This work was supported by the Operation RUMEUR of the PRC C3.

* Corresponding author.

SSDI 0166-218X(93)E0055-4

We consider in this paper MIMD processor networks containing p nodes connected by a configurable interconnection network (which includes any topology of bounded degree) [5].

2. Some preliminaries on broadcasting algorithms

The broadcast routine is very useful for many computations [6, 12]. It consists in sending a message from a given processor to all the others. According to the usual terminology [4], the architecture reference model of the parallel machine that we consider is such that parallel communications are allowed inside a same processor, the links are *full-duplex* (bidirectional) with linear communication time (it takes $\beta + \tau L$ time units to send a message of length L from one processor to any of its direct neighbors where β is the *start-up* and τ the *transmission rate*).

This model corresponds to most of the commercially available machines and in particular the Transputer-based architectures [7]. The routing is managed by software. However, for some new parallel machines the routing is done by special hardware chip, without any intermediate storage between processors [8].

The minimal time to send a message of length L using a pipelined mode is [4]

$$b(L, H) = (\sqrt{(H-1)\beta} + \sqrt{\tau L})^2$$

for an optimal splitting of the initial message into

$$q_{\text{opt}}(L, H) = \left\lceil \sqrt{\frac{\tau L(H-1)}{\beta}} \right\rceil$$

submessages, where H is the length of the chain of processors ($H > 1$).

Obviously, for direct neighbors, the time becomes: $b(L, 1) = \beta + \tau L$. For broadcasting in any networks, we use the previous technique on spanning trees. We obtain the same formula but now, H is the depth of the tree. If Δ arc-disjoint spanning trees can be found, they can be used simultaneously with messages each of size L/Δ, leading to the time:

$$b\left(\frac{L}{\Delta}, H\right) = \left(\sqrt{(H-1)\beta} + \sqrt{\tau\frac{L}{\Delta}}\right)^2. \tag{1}$$

3. Communication model

The target machine that we use for our practical implementations is the *MegaNode* which is designed around the Transputer T800 family [10, 13]. The T800 integrates a high-performance central processing and communication units. The MegaNode architecture is configurable and uses two levels of switching networks (for less than 32

processors, we need only one level of switches to communicate). It can realize any static configurations of degree at most 4.

A precise analysis of the communications of L bytes between 2 neighbor processors of the MegaNode leads to the following expression [9]:

$$\beta_{i,j,k} + \tau_{i,j}L,$$

where subscript i is equal to 0 for the same level of switches and 1 for different levels, subscript j represents the transmission mode (0 for monodirectional 1 for bidirectional), subscript k is the number of parallel processes created for the management of parallel links.

The values that we have measured are given below:

$\beta_{0,0,0} = 11.5$ μs, $\tau_{0,0} = 1.125$ μs/byte,

$\beta_{0,0,k} = 100 + 176k$ μs for $k = 1, 2$ and 3, $\tau_{0,1} = 1.65$ μs/byte,

$\beta_{0,1,k} = 100 + 176k$ μs for $k = 1, 3, 5$ and 7, $\tau_{1,0} = 2.17$ μs/byte,

$\beta_{1,0,0} = 11.4$ μs, $\tau_{1,1} = 2.2$ μs/byte,

$\beta_{1,0,k} = 100 + 176k$ μs for $k = 1, 2$ and 3,

$\beta_{1,1,k} = 100 + 176k$ μs for $k = 1, 3, 5$ and 7.

This model of communication shows that the simple linear model is not valid because the communication parameters (start-up and transmission rate) depend on the network characteristics (depth of the spanning trees and maximum parallelism degree). Usually, when parameters H and $1/\Delta$ decrease, parameters β and τ increase. The two terms of the sum in expression (1) of the broadcast time evolve in the opposite directions:

$$b\left(\frac{L}{\Delta}, H\right) = \left(\sqrt{(H-1)\beta_{i,j,k}} + \sqrt{\tau_{i,j}\frac{L}{\Delta}}\right)^2.$$

Our aim is to find an efficient broadcasting algorithm which achieves the best compromise as possible. Obviously, if the number of processors used for broadcasting is lower than 32, we use a mapping of data with only one level of switches.

4. Description of broadcasting algorithms

For networks with degree lower than 4, the naive strategy consists in sending the whole message (that corresponds to $\Delta = 1$) pipelined on a ternary tree (in Fig. 1 and throughout the paper, the sender node will be represented in grey and for the sake of clarity the orientations of the arcs will be omitted).

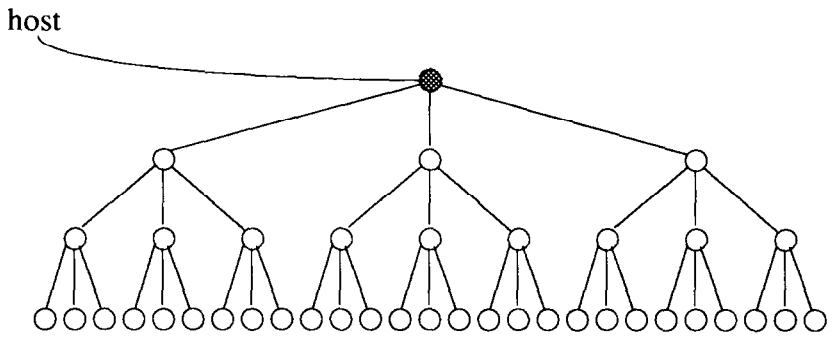

Fig. 1. Ternary tree (40 nodes) of depth 3.

At each time step, a processor receives a packet and sends in parallel 3 times the packet just received at the previous step. The broadcast time is

$$b(L, \lceil \log_3(2p+1) - 1 \rceil) = (\sqrt{(\lceil \log_3(2p+1) - 1 \rceil - 1)\beta_{x,0,3}} + \sqrt{\tau_{x,0}L})^2$$

if $p \leqslant 32$ then $x = 0$ else $x = 1$.

Let us consider now wraparound meshes of p processors. The diameter of a square wraparound mesh is $2\lfloor \sqrt{p}/2 \rfloor$.

We present briefly 4 arc-disjoint spanning trees of minimum depth $2\lfloor \sqrt{p}/2 \rfloor + 1$ using bidirectional links. Each of them is obtained by successive rotations from the others. This result improves the results of Bajwa and Seidel [1] which gave 4 such trees of depth $3\lfloor \sqrt{p}/2 \rfloor$.

We give below the construction of spanning trees of minimum depth for $\sqrt{p} = 4k + 3$ (for any integer k) which is the simplest case because $\lfloor \sqrt{p}/2 \rfloor$ is odd. If $\sqrt{p}$ is even the mesh is not symmetric from the sender point of view [9]. In this case, we have two kinds of basic trees (and we deduce the two others by symmetry). Fig. 2 gives the "East" tree.

The three other trees are obtained by successive rotations over the three other directions (North, West and South). The arc-disjoint spanning trees defined previously are optimal (in the sense of minimal depth) for every size of square wraparound meshes. The proof is straightforward because the depth H of these trees is equal to the diameter plus 1 [9].

Remark. The 4 bidirectional links are used most of the time. In order to have a connection with the host, it is necessary to add an extra processor between the root and the host which does not change the depth of the trees.

The broadcast time is

$$b\left(\frac{L}{4}, 2\left\lfloor \frac{\sqrt{p}}{2} \right\rfloor + 1\right) = \left(\sqrt{2\left\lfloor \frac{\sqrt{p}}{2} \right\rfloor \beta_{x,1,7}} + \sqrt{\tau_{x,1}\frac{L}{4}}\right)^2.$$

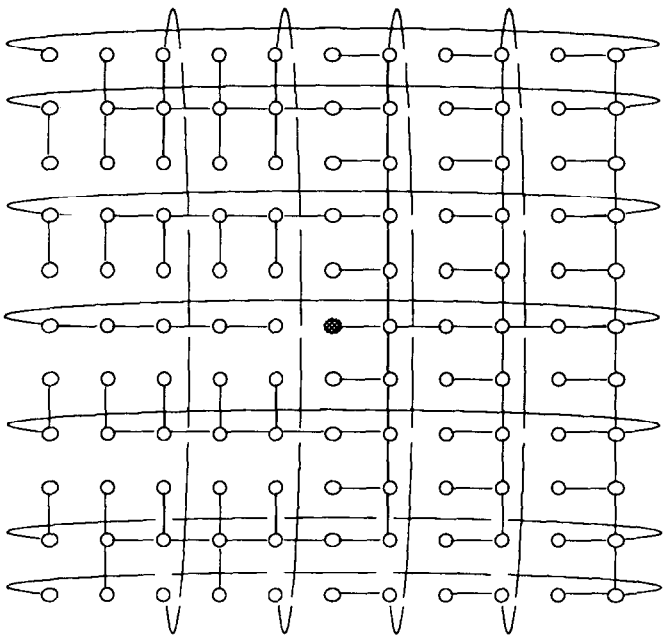

Fig. 2. Minimum depth arc-disjoint spanning tree for a 11×11 bidirectional wraparound mesh.

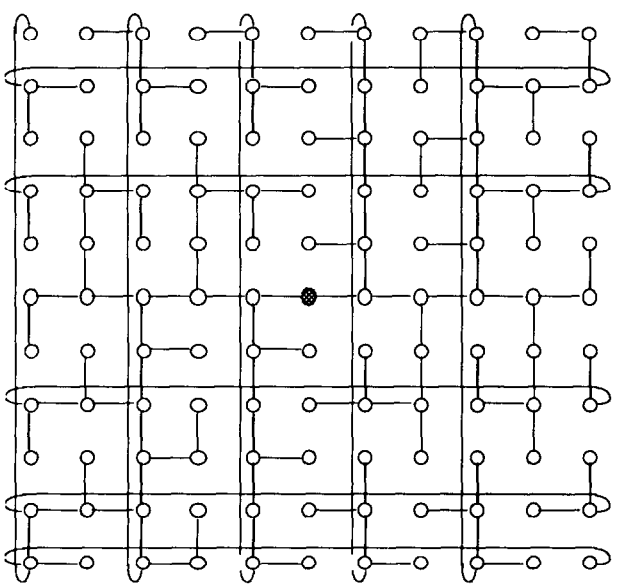

Fig. 3. Minimum depth edge-disjoint spanning tree for a 11×11 monodirectional wraparound mesh.

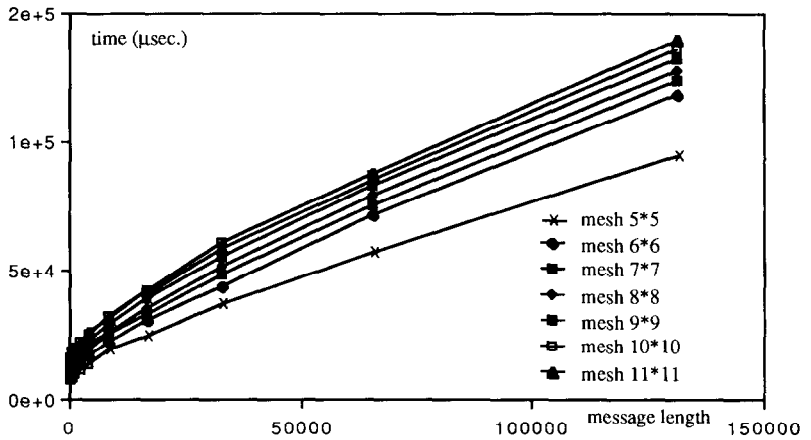

Fig. 4. Broadcasting on bidirectional wraparound meshes.

We can also use monodirectional wraparound meshes. A solution of the broadcasting problem under this assumption has been proposed in [11] and revisited recently by Bermond et al. [3], who propose a solution based on a family of 2 edge-disjoint spanning trees of optimal depth equal to $\sqrt{p}$, so as to pipeline half of the message along each spanning tree. Fig. 3 details the construction of the west–east tree, the other is obtained by rotation). The construction is regular and symmetric, except on the border. We have depicted the construction for meshes of sizes equal to $4k + 3$, the general construction is easy to derive from this basis case [3]:

The broadcast time becomes

$$b\left(\frac{L}{2}, \sqrt{p}\right) = \left(\sqrt{(p-1)\beta_{x,0,3}} + \sqrt{\tau_{x,0}\frac{L}{2}}\right)^2.$$

From [2], we know that for a de Bruijn graph $(2, \log_2(p))$ with bidirectional links there exist 2 arc-disjoint spanning trees of depth $2\log_2(p) + 1$ rooted at any vertex and only $\log_2(p) + 1$ rooted at vertices $(00\ldots0)$ or $(11\ldots1)$:

$$b\left(\frac{L}{2}, 2\log_2(p) + 1\right) = \left(\sqrt{2\log_2(p)\beta_{x,1,5}} + \sqrt{\tau_{x,1}\frac{L}{2}}\right)^2.$$

The binary tree is the simplest structure we can obtain. The broadcast time is

$$b(L, \lceil \log_2(p+1) - 1 \rceil) = (\sqrt{(\lceil \log_2(p+1) - 1 \rceil - 1)\beta_{x,0,2}} + \sqrt{\tau_{x,0}L})^2.$$

5. Implementations

We give now the experimental results corresponding to the broadcasting algorithms. First, we compare the broadcast times for various mesh sizes (Fig. 4).

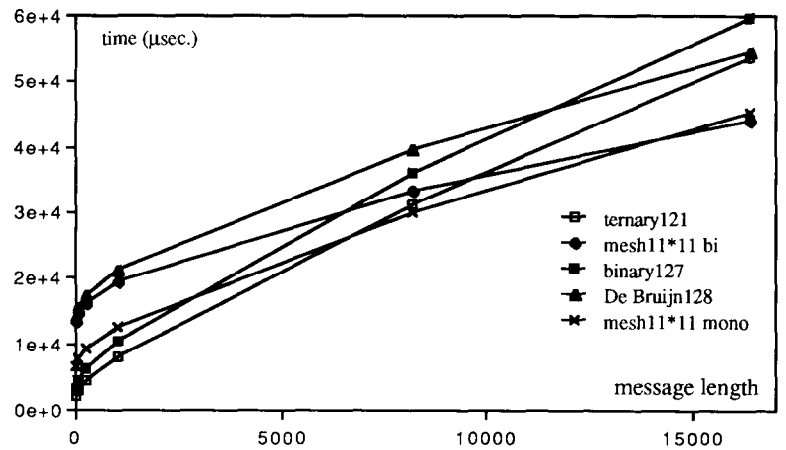

Fig. 5. Comparison of broadcasting strategies for small message length.

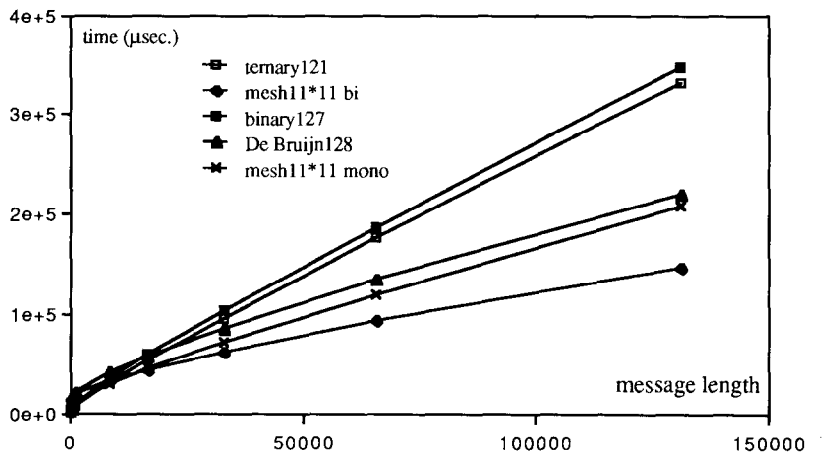

Fig. 6. Comparison of broadcasting strategies for very large messages.

The good relative performances for the 5×5 mesh is due to the fact that it is contained into only one level of switches. Let us now compare the various broadcasts experimentally (Figs. 5 and 6).

6. Conclusion

According to the theoretical model, the experiments show that for very large message sizes, the bidirectional wraparound mesh topology is the best. However, for

message sizes until 7000 bytes the ternary tree is more suitable for broadcasting, then, for message sizes between 7000 and 14 000, the monodirectional mesh is the best.

The modellization that we have presented in this paper emphasizes that the broadcasting algorithms depend significantly on the architecture and that the basic linear model of communication is not sufficient. The main conclusion is that even for medium size messages the simple ternary tree is the best practical topology. However, the solution with arc-disjoint spanning trees on square meshes (for both monodirectional and bidirectional cases) can be very useful for many applications like loading programs.

References

[1] R. Bajwa and S. Seidel, Communication algorithms for tori and grids, Tech. Rept. CS-TR 89-04, Michigan Technological University (1989).
[2] J.C. Bermond and P. Fraigniaud, Broadcasting and gossiping in de bruijn Networks, Tech. Rept. LIP-ENSL RR-9104 (1991).
[3] J.-C. Bermond, P. Michallon and D. Trystram, Broadcasting in wraparound meshes with parallel monodirectional links, Parallel Comput. 18 (1992).
[4] P. Fraigniaud and E. Lazard, Methods and problems of communication in usual networks, Discrete Appl. Math. 53 (1994) 79–133.
[5] K. Hwang and F.A. Briggs, Computer Architecture and Parallel Processing (McGraw-Hill, New York, 1984).
[6] C.-T. Ho and L. Johnsson, Optimum broadcasting and personalized communication in hypercubes, IEEE Trans. Comput. 38 (1989).
[7] Transputer Reference Manual (Prentice-Hall, Inmos Limited, 1988).
[8] P. Kermani and L. Kleinrock, Virtual cut-through: a new computer communication switching technique, Comput. Networks 3 (1979).
[9] P. Michallon, D. Trystram and G. Villard, Optimal broadcasting algorithms on torus, Tech. Rept. RR 8721 LMC-IMAG Grenoble (1991).
[10] D. Nicole, Esprit project 1085 reconfigurable transputer processor architecture, in: Proceedings of CONPAR, Manchester (1988).
[11] Y. Saad and M. Schultz, Data communication in parallel architectures, Parallel Comput. 11 (1989).
[12] Q. Stout and B. Wagar, Intensive hypercube communication. Prearranged Communication in link-Bound Machines, J. Parallel Distrib. Comput. 10 (1990).
[13] T-Node User Manual, Telmat Informatique ref. TN/doc/2-01/3.1 (1990).
[14] E.A. Varvarigos and D.P. Bertsekas, Optimal communication algorithms for isotropic tasks in hypercubes and wraparound meshes, LReport LISD-P-1972 (1990).